

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Absolvování individuální odborné praxe**  
**Individual Professional Practise in the Company**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne.....

podpis.....

## **Abstrakt**

Tato práce popisuje moje působení na individuální odborné praxi ve firmě Control spol. s r. o. v Novém Jičíně, kde jsem po celou dobu trvání této praxe působil jako programátor v programovacím jazyce Java. Dále tato práce obsahuje podrobný popis jednotlivých úkolů zadaných v průběhu odborné praxe a zvolený postup jejich řešení. Cílem této práce je také popsat odborné zaměření této firmy, teoretické a praktické znalosti a dovednosti získané v průběhu studia a uplatněné v průběhu odborné praxe a znalosti a dovednosti scházející v průběhu odborné praxe. Poslední část této práce obsahuje popis dosažených výsledků v průběhu odborné praxe a jejich celkové zhodnocení.

## **Klíčová slova**

Java, Sybase, NetBeans, Java RMI, XML, EDI

## **Abstract**

This work describes my effectiveness at the individual professional practice at Control, limited liability company in Nový Jičín where I was working, during the whole period of my practice, as a programmer in Java programming language. This work includes a detailed specification of individual tasks submitted during the professional practice and selected processes of how they were solved. The aim of this work is also to depict the professional orientation of this company, theoretical and practical knowledge and skills gained within my studies and put into effect within my professional practice and knowledge and skills missing at the professional practice. Last part of this work includes the description of achieved results during my professional practice and their overall evaluation.

## **Keywords**

Java, Sybase, NetBeans, Java RMI, XML, EDI

## **Seznam použitých symbolů a zkratek**

XML - eXtensible Markup Language

SQL – Structured Query Language

TCP/IP – Transmission Control Protocol / Internet Protocol

Java RMI – Java Remote Method Invocation

EDI – Electronic Data Interchange

XSD – Xml Schema Definition

DTD – Document Type Definition

JAXP – Java API for XML Processing

SFTP – Secure File Transfer Protocol

API – Application Programming Interface

DOM – Dokument Object Model

# Obsah

1 Úvod.....	1
1.1 Popis odborného zaměření firmy .....	1
1.2 Pracovní zařazení ve firmě .....	1
2 Zadané úkoly během odborné praxe .....	2
2.1 Komunikace databáze Sybase s prostředím Javy .....	2
2.2 Vzdálené volání metod pomocí Java RMI .....	2
2.3 Java a XML .....	2
3 Zvolený postup při řešení zadaných úloh.....	3
3.1 Komunikace databáze Sybase a Javy .....	3
3.2 Java RMI .....	3
3.3 Java a XML .....	4
4 Teoretické a praktické znalosti a dovednosti uplatněné v průběhu odborné praxe .....	5
5 Znalosti a dovednosti scházející v průběhu odborné praxe.....	6
6 Závěr .....	7
6.1 Dosažené výsledky v průběhu odborné praxe .....	7
6.2 Celkové zhodnocení .....	7
7 Použité zdroje.....	8

# 1 Úvod

## 1.1 Popis odborného zaměření firmy

Firma, ve které jsem absolvoval individuální odbornou praxi se jmenuje Control spol. s r. o. a sídlí v Novém Jičíně. Tato firma se již od svého počátku specializuje na prodej, implementaci a customizaci<sup>1</sup> komplexního informačního systému Dialog. Tento informační systém vychází z úspěšné řady Dialog 2000C a jeho nynější označení verze je Dialog 3000S.

Tento informační systém pokrývá požadavky středních a větších firem na komplexní informační systém pro zpracování informací pro vedení účetní evidence a pro rozhodování v oblasti financí, podpory prodeje a výroby. Informační systém Dialog 3000S nabízí jednotlivé licenčně oddělené moduly řešící logicky oddělené agendy. Tyto moduly mají jednotné uživatelské prostředí a možnost vytváření výstupů s využitím všech existujících dat.

Při vývoji podnikového informačního systému Dialog 3000S bylo využito vývojového prostředí a databázové technologie firmy SYBASE, konkrétně pak nástroje PowerBuilder Enterprise, Infomaker a PowerDesigner. Z databázových aplikací jsou využívány produkty Adaptive Server Enterprise a SQL Anywhere Studio. O některých produktech firmy Sybase bude dále v textu zmínka, protože jsem s nimi při konání odborné praxe pracoval.

## 1.2 Pracovní zařazení ve firmě

Na počátku odborné praxe jsem byl krátce seznámen s tímto informačním systémem a s databázovou technologií, se kterou tento informační systém pracuje. Po svém nástupu na praxi jsem byl zařazen do oddělení vývoje, kde se implementoval, aktualizoval a testoval tento informační systém.

V období mého nástupu na odbornou praxi se ale ve firmě mluvilo o propojení zaběhlé technologie firmy Sybase a moderního objektově orientovaného programovacího jazyka Java. Tato vize se stala mojí hlavní náplní na praxi. Díky těsné integraci technologie Sybase a programovacího jazyka Java, rozsáhlé programátorské dokumentace na internetu a mnoha teoretických a praktických znalostí získaných v průběhu studia šlo o jednoduchý úkol, který dále přerostl ve vývoj aplikací využívajících databázi již zmíněné firmy Sybase.

Mojí hlavní činností po dobu absolvování odborné praxe bylo tedy programování aplikací v jazyce Java a ve vývojovém prostředí NetBeans, práce s relační databází firmy Sybase a po většinu času taky práce s jazykem XML. K práci mi byl přidělen služební Notebook, na kterém bylo nainstalováno všechno potřebné softwarové vybavení.

---

<sup>1</sup> customizace – individuální úprava na míru zákazníka. V tomto případě úprava softwaru.

## 2 Zadané úkoly během odborné praxe

### 2.1 Komunikace databáze Sybase s prostředím Javy

Prvním zadaným úkolem bylo implementovat aplikaci, která by umožňovala po zadání potřebných údajů vytvořit připojení k databázi Sybase z prostředí Javy. Takle aplikace se později měla upravit v obecné rozhraní, které by mohlo být použito v dalších aplikacích implementovaných v Javě a využívající připojení k databázi. Tato aplikace měla být obalena grafickým uživatelským rozhraním pro snazší ovladatelnost. Po úspěšné implementaci měla tato úloha další rozšíření:

- vypisování chybových a varovných hlášek na obrazovku, v případě přerušení spojení, zadání neplatných údajů atd.,
- provádění SQL příkazů nad databází a jejich následné zpracování do čitelné podoby,
- volání uložených procedur s parametrem i bez parametru.

### 2.2 Vzdálené volání metod pomocí Java RMI

Druhým zadaným úkolem bylo navrhnout a implementovat aplikaci, která by využívala rozhraní Java RMI – vzdálené volání metod. Princip této aplikace spočíval v rozdělení na 2 hlavní části – server a klient, kdy klient by mohl vzdáleně přes protokol TCP/IP volat metody, které byly implementovány na aplikačním serveru. Tyto metody mohly být s parametrem i bez parametru a mohly mít různé návratové hodnoty. Hlavní myšlenkou této aplikace bylo provádět všechny SQL dotazy, výpočty a operace s daty na aplikačním serveru a ke klientovi pouze posílat přes TCP/IP protokol požadovaná a již zpracovaná data.

### 2.3 Java a XML

Po splnění předchozích dvou menších úkolů jsem se po zbytek odborné praxe věnoval jazyku XML. Po pečlivém seznámení s touto problematikou a spojením s programovacím jazykem Java, jakožto velice mocným nástrojem pro práci s XML dokumenty, jsem implementoval dvě větší aplikace, které byly později nasazeny do ostrého provozu dvou firem, jedné na Slovensku a druhé v České republice.

První aplikace obsahovala grafické rozhraní pro výpočet a vytváření měsíčních výkazů pojištění a příspěvků a registračních listů fyzických osob. Tyto vytvořené výkazy a registrační listy byly poté exportovány do XML souboru, které se dále elektronicky posílaly do Sociální pojišťovny na Slovensku. Druhá aplikace obsahovala opět grafické rozhraní pro vytváření elektronických objednávek přepravy zboží. Tyto objednávky byly opět exportovány do XML souboru a v tomto formátu byly elektronicky přeneseny (přes protokol SFTP) do jiného informačního systému, kde byly zpracovány. Tomuto systému se říká EDI<sup>2</sup>.

---

<sup>2</sup> EDI – Elektronická výměna dat. Je to moderní způsob komunikace mezi dvěma nezávislými subjekty, při které dochází k výměně standardních strukturovaných obchodních a jiných dokumentů elektronickou formou.

## 3 Zvolený postup při řešení zadaných úloh

Jak již bylo zmíněno výše, všechny vytvořené aplikace byly implementovány v programovacím jazyce Java a ve vývojovém prostředí NetBeans. Tyto aplikace využívaly databázový server Adaptive Server Enterprise, který je produktem již zmíněné firmy Sybase. Abych mohl po implementaci ověřit funkčnost všech vytvořených aplikací, byla mi firmou pro testování poskytnuta testovací databáze.

K tomu, abych mohl vytvořit již samotné připojení k databázi jsem potřeboval důležitou a nepostradatelnou implementaci balíku „jConnect“, který obsahuje potřebný ovladač pro připojení k databázi Sybase. Tento balík byl součástí instalace databázového serveru. Poslední věc, která byla potřeba pro připojení k databázi byl řetězec, který musel být ve správně zadaném formátu. Řetězec obsahoval mimo jiné i adresu serveru, číslo portu a název databáze. Správný formát řetězce jsem si dohledal v dokumentaci, která byla součástí tohoto balíku. Všechny aplikace, které by se implementovaly do budoucna, měly při své činnosti využívat databázi, proto bylo za potřebí vytvořit jednotnou obecnou třídu, tzv. API neboli rozhraní, které by tohle umožňovalo, bylo použitelné pro všechny aplikace a obsahovalo všechny potřebné metody. Toto vytvořené rozhraní se stalo základním stavebním prvkem při implementaci všech dalších aplikací. Při konání odborné praxe jsem čerpal převážně z těchto zdrojů [1, 2, 3].

### 3.1 Komunikace databáze Sybase a Javy

Po vytvoření obecného rozhraní pro připojení k databázi byla část prvního úkolu hotová. Ostatní části již potom nebyl problém implementovat, protože Java má dobře zpracovanou dokumentaci spolu s ukázkami zdrojových kódů a je dostupná na internetu. Bylo pouze třeba zajistit odchycení a ošetření výjimek v případě chyby v SQL dotazu, popř. jiných uživatelských chyb. Tyto vytvořené části aplikace už bylo jenom potřeba obalit do grafického uživatelského rozhraní. Díky vývojovému prostředí NetBeans nebyl v téhle části žádný problém. Při implementaci jsem využil třídy z balíku *java.sql*.

### 3.2 Java RMI

Při řešení druhého úkolu bylo potřeba nejprve zjistit, co to vlastně Java RMI je a jak vlastně funguje. Jedná se o rozhraní Javy, které umožňuje, aby metody vzdálených Java objektů byly volány z jiného JVM<sup>3</sup> na jednom počítači nebo kdekoli v síti. Tato úloha měla sloužit jako server, na kterém byly uloženy všechny metody. Server měl umožňovat připojení určitého počtu klientů, kteří tyto metody mohli přes protokol TCP/IP vzdáleně volat. Tento úkol jsem implementoval jen z části, konkrétně jen tu část, která využívá rozhraní RMI. S tímto úkolem nebyly žádné problémy při implementaci. Při implementaci jsem využil třídy z balíku *java.rmi*.

---

<sup>3</sup> JVM – Java Virtual Machine je sada počítačových programů a datových struktur, která využívá modul virtuálního stroje ke spuštění dalších počítačových programů a skriptů vytvořených v jazyce Java



### 3.3 Java a XML

Poslednímu úkolu, kterému jsem se na odborné praxi věnoval většinu času, byl značkovací jazyk XML a jeho propojení s programovacím jazykem Java. XML jazyk má v dnešní době širší spektrum využití, mě ale zajímal pouze z hlediska výměny dat mezi aplikacemi. Prvním krokem bylo dokonalé seznámení s problematikou XML jazyka a jeho spojením s programovacím jazykem Java. Nasnadě bylo použít JAXP. JAXP je sada několika aplikačních rozhraní vybraných jako základní a nejdůležitější pro zpracování XML v Javě. Za použití vytvořeného rozhraní z prvního úkolu umožňující připojení k databázi a s JAXP bylo možno vytvořit plnohodnotnou aplikaci pro výměnu dat. Tomuhle získávání informací jsem věnoval určitý čas a za použití odborné literatury a internetu jsem se dopracoval k prvním úspěchům.

Prvním krokem ke zpracování XML dokumentu je jeho analýza, kontrola syntaxe, ověření, že dokument odpovídá danému DTD, popř. doplnění neuvedených hodnot atributů z DTD specifikace. Dokument je potom převeden na interní datovou reprezentaci, kterou je možno dále zpracovat. Veškerou tuto činnost zajistí XML parser. V Javě existují dva hlavní přístupy při parsování XML dokumentu:

- přístup založený na stromové struktuře
- přístup založený na událostech.

Přístup založený na událostech jsem nepoužíval, proto se o něm již nebudu dále zmiňovat. U přístupu založeném na stromové struktuře je na základě načítaných dat je vytvořena stromová struktura objektů, které reprezentují původní dokument. Tato struktura pak slouží k dalšímu zpracování. Lze se v ní jednoduše pohybovat z uzlu do uzlu, provádět výpočty, přidávat nebo rušit uzly a atributy. Java podporuje stromovou strukturu DOM, která se nachází v balíku *org.w3c.dom*.

Ovšem v mém případě nebyla potřeba dokumenty číst, ale vytvářet. K tomuto účelu lze tento přístup také využít. Prvním krokem bylo vytvoření struktury XML dokumentu v paměti. Tato struktura byla vždy předem daná. Dalším krokem bylo provést SQL dotaz na databázi, která vrátila požadovaná data. Tato data bylo ovšem nutno upravit, provést výpočty, popř. jinak zpracovat a až poté mohly být vloženy do připravené stromové struktury. Nakonec nezbývá nic jiného, než provést transformaci XML dokumentu (v našem případě jde o DOM dokument v paměti) pomocí vhodných nástrojů. Tyto nástroje se nachází v balíku *javax.xml.transform*. Při transformaci se dokument uloží do souboru na pevný disk. Bylo ovšem ještě potřeba zajistit, aby části XML dokumentu, který byly povinné, byly opravdu vyplněny. Mohlo se stát například, že některé položky v databázi nebyly vyplněny a to se promítlo do našeho XML dokumentu. K tomuto účelu slouží šablony. Jde o jakýsi předpis, jak by měl XML dokument vypadat a které části dokumentu jsou povinné a které ne. Java opět obsahuje několik nástrojů, které dokáží XML dokument se šablonou porovnat, odhalit chybu a informovat o tom uživatele. Tyto šablony byly buď ve formátu DTD nebo XSD. Nástroje pro porovnávání XSD šablon s XML dokumentem se nachází v balíku *javax.xml.validation*. Oproti DTD mají výhodu v tom, že se porovnávají s XML dokumentem až po jeho vytvoření. U DTD se porovnává jen při parsování dokumentu.

## 4 Teoretické a praktické znalosti a dovednosti uplatněné v průběhu odborné praxe

Při vykonávání zadaných úkolů na odborné praxi jsem byl schopen využít mnoha znalostí a dovedností, které jsem získal v průběhu studia i mimo něj. Tyto znalosti byly důležité a nepostradatelné při implementaci jednotlivých aplikací. Ze znalostí získaných v průběhu studia a uplatněných na odborné praxi jsou to konkrétně tyto:

- *programování v jazyce Java* – znalost programovacího jazyka Java a vývojového prostředí NetBeans byla pro mě obrovskou výhodou, protože tato činnost byla mojí hlavní náplní na odborné praxi a tedy i nepostradatelná.
- *znalost značkovacího jazyka XML* – znalost jazyka XML a jeho struktury jsem také uplatnil při implementaci dvou větších aplikací. Tyto aplikace načítaly data z databáze, podle potřeby provedly patřičné výpočty a výsledek uložily na disk do souboru ve formátu XML. Znalost tohoto jazyka a šablony DTD byla v těchto případech nezbytná.
- *Znalost relační databáze* – tato znalost byla pro mě také obrovským přínosem. Bez znalosti relačního datového modelu, uložených procedur, triggerů a ostatních pojmů by plnění zadaných úkolů bylo neúspěšné.
- *Jazyk SQL* – v neposlední řadě to byla znalost jazyka SQL. Veškeré dotazy na databázi byly formulovány pomocí jazyka SQL, čili jeho neznalost by opět vedla k neúspěchu.
- *Ostatní* – znalost pojmů z oblasti počítačových sítí, základních funkcí některých protokolů používaných v informačních technologiích a v prostředí internetu bylo také pro mě výhodou.

## **5 Znalosti a dovednosti scházející v průběhu odborné praxe**

Během absolvování odborné praxe jsem se setkal s menšími problémy, u kterých šlo především v první řadě o neznalost syntaxe v programovacím jazyce Java. Šlo většinou o syntaxi těch metod a tříd, se kterými jsem neměl možnost se setkat v průběhu studia. Tento problém vyřešila ve většině případů programátorská dokumentace, v ostatních případech to byl internet.

Další menší problém byl v případě rozsáhlejších SQL dotazů. V průběhu studia jsem měl možnost se pouze setkat s jednoduššími SQL dotazy. Při řešení úloh šlo většinou o rozsáhlé a složité SQL dotazy, proto jsem se v nich hůře orientoval. Ovšem s pomocí kolegů jsem se vždy dopracoval správného výsledků. Při práci s databází jsem měl zpočátku problém se zorientovat v jednotlivých tabulkách. Celá databáze obsahuje cca 500 tabulek.

Posledním nedostatkem, se kterým jsem se na odborné praxi setkal byla pouze obecná znalost jazyka XML. Tento jazyk se začíná těšit stále větší pozornosti tvůrců i uživatelů informačních technologií. Není divu, neboť podle mnoha předpovědí by se měl záhy stát platformou nového, progresivnějšího stylu práce s elektronickými informacemi. I když problematika jazyka XML je velmi rozsáhlá, přivítal bych jeho podrobnější znalosti.

## **6 Závěr**

### **6.1 Dosažené výsledky v průběhu odborné praxe**

Během absolvování individuální odborné praxe jsem dostal zadáno několik úkolů, z nichž některé byly menšího rozsahu a některé většího. Všechny tyto aplikace se mi povedlo úspěšně naimplementovat. Jak sem již zmiňoval výše, některé aplikace byly nasazeny do provozu ve firmě na Slovensku a v České Republice. Požadavky na tyto aplikace vycházely vždy od zákazníka. Bylo tedy nutné držet se určitých pravidel a požadavků zákazníka. Některé části vývoje aplikací bylo třeba projednat s pověřenou osobou z firmy, pro kterou se implementovala daná aplikace. Byla teda nutná určitá komunikace a obecná znalost z oblasti zaměření dané firmy.

Při výskytu nějakého problému při vývoji některé aplikace bylo vždy potřeba vyhledat osobu, která by daný problém vyřešila, popř. poradila, jak daný problém vyřešit. Díky obětavosti a spolupráci kolegů jsem se časem naučil pracovat samostatně.

### **6.2 Celkové zhodnocení**

Při vykonávání odborné praxe jsem mohl uplatnit v praxi teoretické znalosti získané v průběhu studia a to z učiva z předmětů programování, tvorby informačních systémů, databázových a informačních systémů a teorie zpracování dat. Díky odborné praxe jsem si mohl doplnit některé mezery v učivu a některé věci si na konkrétním příkladu objasnit. Při konání odborné praxe jsem si zároveň dozvěděl i nové věci, které se mi v budoucnu určitě sejdou. Konkrétní příklad je práce s jazykem XML a systémem EDI. Díky této praxi jsem si vytvořil určitý obraz o tom, jak to ve skutečnosti chodí.

Tato odborná praxe byla pro mě obrovským přínosem a jsem rád, že jsem ji mohl absolvovat.

## 7 Použité zdroje

- [1] Sun Developer Network: *Java RMI*  
URL <<http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>> [listopad 2008]
- [2] Sun Developer Network: *Documentation*  
URL <<http://java.sun.com/j2se/1.5.0/docs/api/>>
- [3] Sun Developer Network: *The Java Tutorials*  
URL <<http://java.sun.com/docs/books/tutorial/index.html>>